Mining Hidden Concepts: Using Short Text Clustering and Wikipedia Knowledge

***Abstract*—In recent years, there is a rapid increased use of social networking platforms in the forms of short-text communication. Such communication can be indicative to popular public opinions and may be influential to real-life events. However, due to the short-length of the texts used, the precise meaning and context of such texts are often ambiguous.**

To address these problems, we have devised a new community mining approach that is an adaptation and extension of text clustering using Wikipedia as background knowledge. Based on this method, we are able to achieve high level of precision in identifying the context of communication. Using the same methods, we were also able to efficiently identify hidden concepts from Twitter. Using Wikipedia as background knowledge considerably improved the performance of short text clustering.

I. Introduction

The knowledge insides on-line communities can be very valuable for public options gathering or commercial marketing. Therefore, mining the on-line community in large network such as Twitter becomes one of the most important task in social network analytics (SNA).For example, many candidates use the community mining technique to extract voters' intensions during the election season. It can also be used in recommendation system to group customers with similar interests together.

In traditional community mining approaches mostly are based on statistical SNA and graph theory. They considers the human community and their relationships as a graph. Any node represents an individual, and the edge which connects two nodes indicates the relationship between two individuals in the network. However, the definition of relationship in SNA can only represent that there are some interactions within two users. It is not able to reflect the social relationship of them. For example, the *following/followed* relationships in Twitter are much weaker than *friended* relationships in Facebook.

II. Backgrounds

We first discuss about studies that closely relate to community mining to give readers an overview of this area. We give background knowledges on techniques that have been applied in this thesis and also discuss about their performance.

A. Community Mining

In this section, we discuss about community mining based on social network analysis approaches and based on other interesting approaches to see the overall perspective of community mining.

1) Social Network Analysis Approaches: According to [8], "Social network analysis studies social networks by means of analyzing structural relationships between people". Mining Community using traditional social network analysis approaches usually focuses on structure of the social network that is represented by direct or indirect graph. Each node in the graph represents an instance in the network e.g. person or object whereas links between nodes represent relations between the instances. The relations between the instances in the network can be defined by explicit information such as friends in Facebook or followers in Twitter.

[15] used structure of subgraph in network to identify groups of people that share the same interests based on email history. [8] analyzed structure of network to identify communities among Slashdot users. Method of [8] is based on social network analysis approaches with negative weighted edges graph. [19] used an algorithm based on graph theory to identify signed social networks. [3] mentioned that using minimum cut framework can efficiently identify members in a community. A study of [12][7] also detected communities by considering the network structure. [12] took the network structure properties such as loops and edges of the network into the account. [7] considered bi-partite subgraphs to locate communities of websites.

However, there is a limitation of discover communities that are formed by hidden relations between the instances (e.g. people share the same interests in social network analysis approaches) because these approaches mostly model only explicit relations of the instances in the network [16].

B. Wikipedia Concepts Identification and Disambiguation

Recently, Wikipedia is used in many fields that are related to machine learning such as natural language processing, text classification and text clustering. One of difficulties for using Wikipedia is accurately matching between input text and Wikipedia concepts (articles) because each word or each phase in the input text can refer to one or more Wikipedia concepts. For example, a word "apple" can refer to both concepts "Apple (fruit)" and "Apple Inc.".

This problem has been interesting for a while and a lot of researchers have been proposed many methods to solve this problem. The review of word sense disambiguation methods can be found at [11].

An alternative approach that performs well in doing word sense disambiguation is using machine learning methods to learn labeled training set and classify ambiguous words. This concept is used in text annotation with Wikipedia links. [1][2][9][10] are several researches regarding to text annotation with Wikipedia links. The first paper published about

Wikipedia as a resource for annotation is [9] before significant improvement on this field by [10].

However, most papers did experiment in the context of standard length document. These experiments do not ensure that the approaches used in the papers will perform well in the case of short text document such as tweets, news or search snippets. In 2010, [2] brought the concept of annotating plain-text with Wikipedia links to context of short length document. They used anchor as a resource of identification instead of using only Wikipedia title as [18] because anchors are selected appropriately by people who create the pages. Their approach consists of three main steps: anchor parsing, anchor disambiguation and anchor pruning. Performing those steps extends an ability to deal with short text for the annotation system. Overall performance of [2]'s system improves compared to [10]'s system for both short and long text document cases significantly.

III. Proposed Method

In the case of short text document, low term frequency and many uses of abbreviation are its main characteristics which directly affect the performance of bag-of-word model in clustering task. Low term frequency makes only inverse document frequency term left in TF-IDF model resulting in poor clustering performance. Use abbreviation makes several different representations of the same word. This leads bag-of-word model treats them as different words which is not appropriate. We adopted the idea from short text annotation in [2] and adapted the concept of document enriching strategies from [5] and [18] to use with our project because they already proved their performance with short text documents. With suitable Wikipedia concept identification and disambiguation process for short text document and good document enriching strategies, we are able to improve the performance of document clustering in the case of short length text document. Our approach is a combination of 3 main tasks as follow:

1) Wikipedia Concepts identification. The main responsibility of this part is to identify Wikipedia concepts in the documents.
2) Document Enriching. This subsystem is for enriching the tweets with Wikipedia concepts corresponding to each document getting from above step.
3) Document Clustering. This subsystem identify communities in the documents both enriched documents from step 2 and original documents using text clustering method.

It started from pre-processing Wikipedia data dump. We extracted all anchors and links in Wikipedia article and indexed into a catalog. For efficiency, we also indexed all Wikipedia pages, their content and categories into another catalog for efficiency in querying. After that, in the concepts identification process, all tweets in our dataset are searched for their related Wikipedia concepts. Next, each tweet is enriching with 2 different strategies based on their related concepts. We will discuss about this in latter section. Those enriched tweets are stored in the database for efficiency. Finally, all tweets are clustered with Bisecting *k*-means clustering based on similarity of their enriched contents.

A. Wikipedia Concepts identification and Disambiguation

We use an anchor that is a text that used to describe a link between Wikipedia pages as the main resource. Basically, an anchor uses words or phrases to describe a Wikipedia page it links to. An anchor normally uses a title, synonym or acronym of the page. However, it also uses a phrase that may be exactly different from page title. Table 1 shows some examples of anchors and their corresponding Wikipedia concepts. Using anchor text, we are able to identify Wikipedia concepts in the document not only by the title of Wikipedia concepts but also synonyms, acronyms or phrases that refer to those concepts as well.

Nevertheless, each anchor often refer to two or more than two Wikipedia concepts. Thus, we need word sense disambiguation process to select the most appropriate page that referred by an anchor. Hence, in this Wikipedia concepts identification process, we can split into four sub-processes.

1) Preprocessing Wikipedia: We use 4th July 2012 English Wikipedia article dump which contains 4,012,083 articles and has a size about 8.2GB compressed. Then, we preprocessed and indexed them into 2 main catalogs to speed up the query.

1) Anchor Dictionary. We extracted all links and their anchors in Wikipedia pages and built them as anchor dictionary. The anchor dictionary is not like English dictionary. It contains only two important information: 1) anchors and 2) their corresponding Wikipedia concepts.
2) Wikipedia Pages. We also indexed Wikipedia pages content, their categories and inlink for speed in querying.

2) Anchors Identification: In order to identify Wikipedia concepts related to each tweet, we need to identify all anchors appearing in the tweet. In this sub-process, we use the steps given in Algorithm 1 to find the anchors. $lp(a)$ is link probability that can be calculate by following equation:

$$lp(a) = \frac{link(a)}{freq(a)} \tag{1}$$

where $link(a)$is number of anchor a used as a link and $freq(a)$ is numberof anchor a appearing in all documents in collection.

Algorithm 1 Document Parsing

Require: input document d
  A =ngrams(d, n=6)
  for each $word \in A$ **do**
    if $word \notin dictionary$ **then**
      A=A$\{word\}$
    end if
  end for
  for $a_1 \in A$ **do**
    for $a_2 \in A$ **do**
      if $a_1 \neq a_2$ and $substring(a_1, a_2)$ and $lp(a_1) < lp(a_2)$ **then**
        A=A$\{a_1\}$
      end if
    end for
  end for

3) Concepts Disambiguation: Each anchor in set of candidate anchors we get from previous section can refer to several Wikipedia concepts. Therefore, in disambiguation step, we disambiguate those concepts and assign the most appropriate concept to each anchor.

The same anchor may have different meanings and may link to different Wikipedia concepts depending on the context of the document. Therefore, we need disambiguation process in order to select the appropriate Wikipedia concepts. We used voting scheme adopted from [2]. The idea behind this voting scheme is that every anchor has to vote all Wikipedia concepts related to other anchors in the document except concepts related to itself. Then, the concepts that are in top-e rank considering from voting score will be selected as candidate concepts. Finally, we select the most appropriate concept by using commonness score.

Using only this score to assign the concepts to the anchors may not be enough because, as mentioned in [10], balancing between the score and commonness is the main factor affecting the performance. In our case, computational efficiency is our main concerns because we need to process a lot of tweets. Therefore, we adopt only disambiguation by threshold method to filter out unrelated concepts. The following is the steps to perform disambiguation by threshold:

1) Remove all concepts that have $rel_a(p_a) < \delta$, where $\delta = 0.3$
2) Remove all concepts that have
$$\frac{rel_a(p_{top\_a}) - rel_a(p_a)}{rel_a(p_{top\_a})} > \epsilon \quad (2)$$
in which p_{top_a} is a concept corresponding to anchor a getting highest rel score and $\epsilon = 0.30$
3) Finally, the concept that has highest commonness $Pr(p_a|a)$ is assigned to an anchor a

4) Concepts Filtering: However, after disambiguation step is applied, there may be uncorrelated concepts left. Therefore, we have concepts filtering step to remove all concepts that are not related to others. We filter out unrelated anchors by using the concept of coherence between selected concepts from concepts disambiguation step. To calculate coherence score, we use average relatedness between selected concepts as follow:

$$coherence(a \rightarrow p_a) = \frac{1}{\|S\| - 1} \sum_{p_b \in S\{p_a\}} rel(p_a, p_b) \quad (3)$$

where S is number of all selected concepts. Then, we filter out unrelated anchors that are satisfy this following condition:

$$\frac{coherence(a \rightarrow p_a) + lp(a)}{2} < \epsilon \quad (4)$$

where $\epsilon = 0.2$ and $lp(a)$ is link probability of an anchor a.

B. Document Enriching

A tweet is in a kind of very short text document which has very low term frequency and usually consist of only important words. The reason is that character limitation of tweet that users can compose is only 144 characters or about 10 words. Therefore, most terms in the tweets tend to appear only once. Moreover, due to the same reason, to express everything the users think, they need to select only the important words that enough for expressing all information they want to communicate.

With these characteristics of short text document, there are some problems with TF-IDF weighting that we use for modeling the document. First, low term frequency in each document results in there is only inverse document frequency term left. Second, most tweets tend to have only important words. This means, in some cases, important words will have lower inverse document frequency compared to those who are not important. This means, in some cases, important words will have lower inverse document frequency compared to those who are not important.

[5] succeed in using 3 different strategies to enrich TF-IDF vector with background knowledge based on WordNet. Three strategies consist of adding corresponding WordNet concepts, replacing terms by WordNet concepts and replacing term vector with concept vector. Also, [18] yielded the good results from enriching the document with semantic related terms based on Wikipedia knowledge. In our proposed method, we adapted the strategies from [5] and [18] to enrich Wikipedia knowledge into the tweets.

1) Strategy 1: Add Wikipedia concepts: We replace and add terms in each tweet with its corresponding Wikipedia concepts. The reason is that one of the problems that reduce the performance of bag-of-word model is that each Wikipedia concept can be mentioned by several different words/phrases. For example, there are many words/phrases that used to refer to concept "Microsoft" such as "microsoft corp.", "ms", "microsoft corporation" and "microsoft".

This problem leads to low cosine similarity between these two documents because the word "MS" and "Microsoft" are treated as different words. Therefore, to reduce the error of different representation of the same concept, we replace them with Wikipedia concept which change them into the same representation.

Moreover, as we mention earlier about problems of TF-IDF weighting, we also add related Wikipedia concepts into the tweets in order to make important terms have higher score.

2) Strategy 2: Add Wikipedia concepts and categories: We extend the first strategy by adding categories of Wikipedia concepts corresponding to each term in a tweet because another problem of bag-of-word model is that the model cannot capture semantic relationship between two related terms. Adding Wikipedia categories will solve the problem of semantic relationship between the tweets. As an example, for ease in understanding, given two tweets that has only one term about Google services as follow: "Gmail" and "Youtube". The cosine similarity between two terms is 0. But, if we add "Google Service" which is one of common categories between these two terms, we will get cosine similarity more than 0. Adding Wikipedia categories can solve the problem semantic relationship of bag-of-word model. Therefore, in this strategy, we also add Wikipedia concepts in documents besides replacing and adding Wikipedia concepts in Strategy 1.

C. Document Clustering

After the tweets are enriched with Wikipedia knowledge, in this step, we mine communities from these tweets by clustering them into groups based on their topics. However, before we apply clustering algorithm, we need to preprocess the enriched tweets resulting from applying two strategies in the previous section into TF-IDF vectors.

1) Preprocessing Twitter Data:

1) *Tweet Filtering*. Due to some of tweets in our dataset do not contains any useful information, which are outliers that can be reduce the performance of document clustering process.
2) *Stop Words Filtering*. Removing stop words helps improve the performance of the model for clustering and classification task. Mostly, stop words are short function words such as pronouns, prepositions and conjunction. We used NLTK stops word list and also added some extra stop words that mostly occur only in Twitter such as "lol", "huh" into the list.
3) *Word Stemming*. Stemming is a method that aim to reduce a word into its root form. The effects of stemming in text clustering and TF-IDF model are shown in [6] One of the key advantages of stemming is that they reduce the dictionary size. Moreover, it makes us able to match the same word with its different form. In this paper, we used Porter Stemmer [13] which is the stemming algorithm that was invented by Martin Porter.
4) *Tweet Dictionary*. We built dictionary of the words that appearing in the tweets after stemming and filtered out words that occur less than 5 times and words that occur in the tweets more than half proportion of all tweets in the dataset.
5) *TF-IDF Vectors* After we processed all of the tweets following 4 steps above. We convert contents of the tweets into TF-IDF vectors by using equation 2.

2) Bisecting K-means Clustering: The results from many papers show that affinity propagation is the best among several clustering algorithms in short text clustering task. However, there are many criticism of affinity propagation about a problem with a large dataset. The issue about scalability of affinity propagation was mentioned in [4], [20]. Therefore, due to the size of our dataset, we decided to use Bisecting k-means clustering because of its scalability and efficiency. The detail of the algorithm are shown in Algorithm 2.

Algorithm 2 Bisecting K-means clustering

repeat
  Select a cluster from the list of clusters
  for $i = 1$ *to* $number\_of\_iterations$ **do**
    Bisect the selected cluster using basic k-means
  end for
  Take the split that produces the clustering with the highest overall similarity
until the list of cluster contains K clusters

IV. Evaluation

The aim of our project is mining community in Twitter by finding a group of tweets which have the same concepts. Our approaches are based on text clustering and integration of Wikipedia knowledge and vector space model.

A. Experiments

In order to evaluate our methods, we did three experiments as follow:

- *[Experiment 1]* In order to evaluate our work, we need some approaches to be compared with. Therefore, before experiment on our approaches, we ran experiment on pure clustering algorithm without enriching Wikipedia knowledge. First, we preprocessed around a million tweets and model them with IF-IDF vectors. Then, we clustered them directly without any further preprocessing.
- *[Experiment 2]* In this method, we did further preprocessing step with our tweets data by adding related Wikipedia concepts as we explained in Strategy 1. After that, we model those enriched tweets with TF-IDF vectors before clustering.
- *[Experiment 3]* This method extended the concepts from Method 2. It does not only add Wikipedia concepts that related to each tweets in the preprocessing step but also add Wikipedia categories of each Wikipedia concepts into the tweets in order to solve the samantic relation problem of bag-of-word model. Then, these preprocessed tweets are modelled using TF-IDF vectors. Finally, we cluster it with these enriched tweets in the same way as the two previous methods.

B. Evaluation based on testset

For evaluation based on testset, we manually labelled 400 tweets with appropriate groups. Then, we evaluate all three methods we mentioned earlier by calculating V-Measure score between true labels and labels that were assigned by clustering algorithm. In this section, we first describe about testset we used and then explain the detail of evaluation metric (V-Measure) in the later subsection.

1) Dataset: For the first step, to set up the clustering algorithm, we need to identify number of clusters. We manually selected 400 tweets from 20 groups as a testset. Then, we ran clustering algorithm repeatedly with different number of clusters on this testset to find the most appropriate number of topics.

2) Evaluation Metric: Typically, the basic criteria of a clustering result are homogeneity and completeness. The homogeneity criterion is satisfied, for all clusters, every member of each cluster comes from only one class which is defined as:

$$homogeneity = \begin{cases} 1 & \text{if } H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{otherwise} \end{cases} \quad (5)$$

where $H(C|K)$ is the conditional entropy of the classes given assigned clusters and $H(C)$ is the entropy of the class defined

as:

$$H(C|K) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{n_{c,K}}{n}\log\frac{n_{c,k}}{\sum_{c=1}^{|C|}n_{c,k}} \quad (6)$$

$$H(C) = -\sum_{c=1}^{|C|}\frac{\sum_{k=1}^{|K|}n_{c,k}}{n}\log\frac{\sum_{k=1}^{|K|}n_{c,k}}{n} \quad (7)$$

in which n is number of all data points and $n_{c,k}$ is number of data points from class c that clustered into cluster k.

Completeness criterion is quite opposite to homogeneity. It is satisfied if all members of a class are clustered into the same cluster. Mathematically, we can define completeness score as follow:

$$completeness = \begin{cases} 1 & \text{if } H(C,K) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{otherwise} \end{cases} \quad (8)$$

where $H(K|C)$ is the conditional entropy of assigned clusters given the classes and $H(K)$ is the entropy of assigned clusters defined as:

$$H(K|C) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{n_{c,K}}{n}\log\frac{n_{c,k}}{\sum_{k=1}^{|K|}n_{c,k}} \quad (9)$$

$$H(K) = -\sum_{k=1}^{|K|}\frac{\sum_{c=1}^{|C|}n_{c,k}}{n}\log\frac{\sum_{c=1}^{|C|}n_{c,k}}{n} \quad (10)$$

in which n is number of all data points and $n_{c,k}$ is number of data points from class c that clustered into cluster k.

A good clustering result should satisfy both homogeneity and completeness at the same time. In order to do that, we used V-Measure as a metric for evaluating clustering results. V-Measure which is first introduced by [14] is the harmonic mean of homogeneity and completeness scores bounded in the range of [0, 1] . The closer the value is to 1, the better the quality of a clustering result. It can be defined as the following equation:

$$V_\beta = \frac{(1+\beta) * h * c}{(\beta + h) + c} \quad (11)$$

where β is the weight, if β is set less than 1, homogeneity is weighted more. If β sets to more than 1, completeness is weighted more. In our experiment, we weight them equally by setting $\beta = 1$.

C. Results

For each experiment, we ran bisecting k-means multiple times with different setting up of number of clusters k ranging from 64 to 4096. Then, we evaluate the clustering results with our testset using V-Measure as an evaluation metric. Figure 1 shows V-Measure scores of Experiment 1, 2 and 3 with different setting of number of clusters k.

From the figure, it is clear to see that *Method 2 (concepts)* and *Method 3 (concepts+categories)* outperformed *Method 1 (baseline)*. *Method 2* is clearly better than *Method 1* at every setting of number of clusters k. The highest V-Measure that *Method 1* can get is 0.674 at $k = 3800$ whereas the highest V-Measure of *Method 2* is 0.747 at $k = 3400$. The difference between the highest peak of them is 7.3%.

Furthermore, in the case of *Method 3*, it has clearly higher performance than *Method 1* in the Figure 1. Comparing with their best performance, *Method 3* gets 14.7% better with V-Measure 0.821 at $k = 3600$. We can conclude from these results getting from the testset that *Method 2* and *Method 3* have dramatic improvement from *Method 1* with V-Measure 0.747, 0.821 and 0.674. From these results, it confirms that using Wikipedia as a resource for enriching the tweets can improve the performance of community mining.

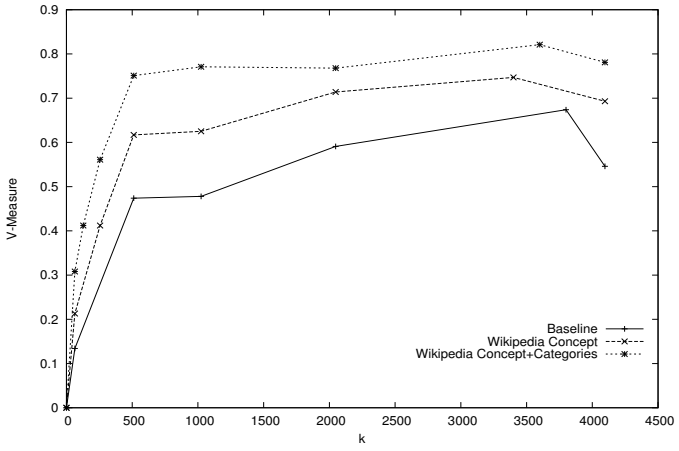

Fig. 1. Comparison among Method 1,2 and 3

D. Analysis

There are two main reasons of the improvement of the two approaches that use Wikipedia to enrich the original documents. First, adding Wikipedia concepts has positive effect with TF-IDF model and can overcome the problem of this model having with short text document as we mentioned in earlier sections. Second, the reason that make the third approach got the best results that is adding Wikipedia categories solve one of pitfall of TF-IDF model which is the problem of semantic relatedness between the terms in the documents. As a consequence, the improvement of TF-IDF model results in better clustering performance.

1) Effect of document enrichment on TF-IDF: In general, baseline method cannot reflect the true importance of the terms. Here is an example from the real dataset. n our corpus, the word "twitter" appears in 91,821 tweets whereas the word "random" only appear in 5,947 tweets. It means the word "random" get higher TF-IDF weight than the word "twitter" which is not appropriate. As a consequence, baseline method got the worst results from both our testset and survey because the tweets are represented with inappropriate model.

This improvement of the model results in better quality of clustering process because the clustering algorithm that we used, k-means clustering, tries to cluster the similar tweets together. As a result, with TF-IDF vector of baseline method, k-means will group this tweet into a group that the word "random" is important. But, with TF-IDF vector of other two methods that used Wikipedia knowledge, they will group this tweet into a group that the word "twitter" is important. In other words, k-means clustering tries to assign a tweet into the closest group which is the group whose centriod closest to

the tweet in vector space. As a consequence, the results from methods that enrich the tweets with Wikipedia concepts are significantly better than baseline method.

2) Effect of document enrichment with semantic relationships: The method that enriched documents with Wikipedia categories yielded the best V-Measure score in Figure 1. It has considerable improvement compared to baseline approach and is slightly better than enriching with only Wikipedia concepts. The reasons behind its performance is that enriching with Wikipedia categories solve the problem of semantic relatedness of TF-IDF model. In TF-IDF model, we normally use cosine distance to describe the relatedness between documents. Further distance between two documents means less similarity and lower relatedness. Without adding the categories, cosine distance is not able to reflect relationship between any two semantic related terms. Then, the distance between them is the upper-bound of cosine distance which is 1. However, after adding the categories, the two semantic related terms become closer in TF-IDF vector space with cosine distance.

V. Conclusion

In this paper, we proposed a state-of-the-art method to mine hidden concepts from large scale short text documents based on Wikipedia knowledge. The optimistic result we have is the method enriched with Wikipedia concept and categories. Based on the evaluation metric in Section 4.3, our method has a promising V-measure score up to 0.821 from real Twitter data where baseline method only has 0.674 in V-measure score.

Mining community based on social network analysis approaches fails to capture hidden concepts in a network because they usually model the network as a graph with explicit relationships that can be found in the network connectivity. The hidden concepts in the network are not taken into account leading these approaches to missing the important concepts among users in the network. In this study, we therefore investigated an alternative approach to identify hidden concepts in Twitter. We used clustering based methods to mine hidden concepts in tweets based on their topics.

References

[1] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *In Proc. 2007 Joint Conference on EMNLP and CNLL*, pages 708–716, 2007.

[2] P. Ferragina and U. Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1625–1628, New York, NY, USA, 2010. ACM.

[3] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 150–160, New York, NY, USA, 2000. ACM.

[4] Y. Fujiwara, G. Irie, and T. Kitahara. Fast algorithm for affinity propagation. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2238–2243. AAAI Press, 2011.

[5] A. Hotho, A. Hotho, S. Staab, S. Staab, G. Stumme, and G. Stumme. Text clustering based on background knowledge, 2003.

[6] M. Kantrowitz, B. Mohit, and V. Mittal. Stemming and its effects on tfidf ranking (poster session). In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, pages 357–359, New York, NY, USA, 2000. ACM.

[7] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *Proceedings of the eighth international conference on World Wide Web*, WWW '99, pages 1481–1493, New York, NY, USA, 1999. Elsevier North-Holland, Inc.

[8] J. Kunegis, A. Lommatzsch, and C. Bauckhage. The slashdot zoo: mining a social network with negative edges. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 741–750, New York, NY, USA, 2009. ACM.

[9] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *In CIKM ?07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM, 2007.

[10] D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 509–518, New York, NY, USA, 2008. ACM.

[11] R. Navigli. A quick tour of word sense disambiguation, induction and related approaches. In *Proceedings of the 38th international conference on Current Trends in Theory and Practice of Computer Science*, SOFSEM'12, pages 115–129, Berlin, Heidelberg, 2012. Springer-Verlag.

[12] M. E. J. Newman. Detecting community structure in networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 38(2):321–330, Mar. 2004.

[13] M. F. Porter. Readings in information retrieval. chapter An algorithm for suffix stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[14] A. Roseberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure.

[15] M. F. Schwartz and D. C. M. Wood. Discovering shared interests using graph analysis. *Commun. ACM*, 36(8):78–89, Aug. 1993.

[16] D. Shen, J.-T. Sun, Q. Yang, and Z. Chen. Latent friend mining from blog data. In *Data Mining, 2006. ICDM '06. Sixth International Conference on*, pages 552 –561, dec. 2006.

[17] O. Tsur, A. Littman, and A. Rappoport. Scalable multi stage clustering of tagged micro-messages. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 621–622, New York, NY, USA, 2012. ACM.

[18] P. Wang, J. Hu, H.-J. Zeng, and Z. Chen. Using wikipedia knowledge to improve text classification. *Knowl. Inf. Syst.*, 19(3):265–281, May 2009.

[19] B. Yang, W. Cheung, and J. Liu. Community mining from signed social networks. *Knowledge and Data Engineering, IEEE Transactions on*, 19(10):1333 –1348, oct. 2007.

[20] X. Zhang, C. Furtlehner, and M. Sebag. Distributed and incremental clustering based on weighted affinity propagation. In *Proceedings of the 2008 conference on STAIRS 2008: Proceedings of the Fourth Starting AI Researchers' Symposium*, pages 199–210, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press.